Lindstrom Quantifier and Bounded Arithmetic for LOGCFL (Formal Systems and Computality Theory)

AUTHOR(S):

Kuroda, Satoru

CITATION:

Kuroda, Satoru. Lindstrom Quantifier and Bounded Arithmetic for LOGCFL (Formal Systems and Computality Theory). 数理解析研究所講究録 2011, 1729: 67-83

ISSUE DATE:

2011-02

URL:

http://hdl.handle.net/2433/170548

数理解析研究所講究録
第1729巻 2011年 67-83

Lindström Quantifier and Bounded Arithmetic for LOGCFL

Satoru Kuroda
(Gunma Prefectural Women's University)

1 Introduction

In this paper we give a general framework for building bounded arithmetic theories using complete problems for certain complexity classes. As an application we define a bounded arithmetic theory for LOGCFL based on the acyclic conjunctive query problem.

Formulations and techniques used here are based mainly on previous works by Kolokolova [3] and Nguyen [5]. Nevertheless, we treat complexity classes which are seemingly unable to handled directly by their formulations.

This can be illustrated by comparing two complexity classes, namely NL and LOGCFL. These two classes both contains the concept of nondeterminism. However, they have rather different nature.

Consider the st-connectivity problem which is complete for NL. The witness for an instance of it is, of course, a path from the start node s to the goal node t.

On the other hand, it can also be witnessed by all nodes which are reachable from the node s and this can be computed in deterministic polynomial time.

This means that the nondeterminism of NL has a deterministic alternative with high feasibility.

On the contrary, it seems unlikely that LOGCFL also has this property.

Since Nguyen's system **VNL** for NL heavily depends on the property, the above argument suggests that it is hard to construct a similar system for LOGCFL.

So instead we consider another property of when a complexity class as a class of predicates has a nice counterpart in function class.

2 Preliminaries

In this section we give basic concepts on bounded arithmetic and complexity theory.

2.1 Complete problems via AC^0 reductions

First we will give a brief tour of complexity theories within PTIME. The smallest class we consider is the class AC^0 which consists of all predicates decidable by families of constant depth polynomial circuits of unbounded fan-in.

As we will consider bounded arithmetic theories of two sort, we deal with number and string objects in complexity theory as well. Lower case variables $x_0, x_1, \ldots$ refer to numbers while upper case variables $X_0, X_1, \ldots$ refer to strings.

For most classes within PTIME, complete problems are known under AC^0 reductions. Let $\mathcal{C}$ be a complexity class and $\mathcal{K}$ be a complete set of $\mathcal{C}$ under AC^0 reductions. Let $C_{\mathcal{K}}(\vec{x}, \vec{X})$ be the characteristic function of $\mathcal{K}$.

The functional version of the class $\mathcal{C}$ is denoted by $\mathcal{FC}$ and is defined to be the class of functions of polynomial growth which are bitwise computable in $\mathcal{C}$. We shall give a recursion theoretic characterization of $\mathcal{FC}$ which will be used to relate $\mathcal{C}$ to a two-sort theory.

Definition 1 *We define the following initial functions:*

$$Z(x) = 0,\ S_0(X) = X0,\ S_1(X) = X1,\ |X| = \textit{the length of } X,$$
$$x + y,\ x \cdot y,\ MSP(X, y).$$

We use the following extremely weak recursion to characterize AC^0 reductions which was essentially due to Clote and Takeuti:

Definition 2 *A function $F(n, \bar{x}, \bar{X})$ is defined by Concatenation Recursion on Notation (CRN) from $G(\bar{x}, \bar{X})$ and $H(n, \bar{x}, \bar{X})$ if*

$$\begin{aligned} F(0, \bar{x}, \bar{X}) &= G(\bar{x}, \bar{X}), \\ F(n+1, \bar{x}, \bar{X}) &= \begin{cases} S_0(F(n, \bar{x}, \bar{X})) & \text{if } |H(n, \bar{x}, \bar{X})| = 0, \\ S_1(F(n, \bar{x}, \bar{X})) & \text{if } |H(n, \bar{x}, \bar{X})| > 0. \end{cases} \end{aligned}$$

Then we have

Theorem 1 *Let $\mathcal{K}$ be a complete problem of $\mathcal{C}$ under AC^0 reductions and $C_{\mathcal{K}}$ be its characteristic function. If the class $\mathcal{C}$ is closed under AC^0 operation then $\mathcal{FC}$ is the smallest class of functions containing initial functions of Definition 1 together with $C_{\mathcal{K}}$ and closed under composition and CRN operations.*

(Proof). Let $\mathcal{F}_{\mathcal{K}}$ be the closure of INITIAL and $C_{\mathcal{K}}$ under composition and CRN operations. First we prove that $\mathcal{FC} \subseteq \mathcal{F}_{\mathcal{K}}$. Notice that for any $P(\bar{x}, \bar{X}) \in \mathcal{C}$, its characteristic function

$$f_P(\bar{x}, \bar{X}) = \begin{cases} 1 \text{ if } P(\bar{x}, \bar{X}) \\ 0 \text{ otherwise} \end{cases}$$

is in $\mathcal{F}_{\mathcal{K}}$. Then any $F \in \mathcal{FC}$ can be defined by CRN operation using such characteristic functions.

Conversely, we show that any $F \in \mathcal{F}_{\mathcal{K}}$ is in $\mathcal{FC}$ by induction on its recursive definition. It is readily proved that any function in INITIAL is in $\mathcal{FC}$. It is also immediate to see that $C_{\mathcal{K}} \in \mathcal{FC}$.

To prove the closure of $\mathcal{FC}$ under composition, let $F(\bar{x}, \bar{X}, Y), G(\bar{x}, \bar{X}) \in \mathcal{FC}$. It suffices to show that for any $i < |F(\bar{x}, \bar{X}, G(\bar{x}, \bar{X})|$, the ith bit of $F(\bar{x}, \bar{X}), G(\bar{x}, \bar{X})$ can be determined by an algorithm in $\mathcal{C}$. To compute it, we first need the value of $G(\bar{x}, \bar{X})$, that is all bits of $G(\bar{x}, \bar{X})$. Since $\mathcal{C}$ is closed under complementation, we have $\mathcal{C}$-algorithms to decide both $G(\bar{x}, \bar{X})(i)$ and $\neg G(\bar{x}, \bar{X})(i)$. Thus by executing both algorithms simultaneously, for all i, we may know all bits of $G(\bar{x}, \bar{X})$ in $\mathcal{C}$ as it is closed under unbounded fan-in AND.

Finally we show that $\mathcal{FC}$ is closed under CRN operation. Let $G(\bar{x}, \bar{X}), H(n, \bar{x}, \bar{X}) \in \mathcal{FC}$ and $F(n, \bar{x}, \bar{X})$ be defined by CRN from G and H. Let $i < |F(n, \bar{x}, \bar{X})|$. If $i < |G(\bar{x}, \bar{X})|$ then the ith bit of $F(n, \bar{x}, \bar{X}, Y)$ is decided by the algorithm for G which is in $\mathcal{C}$ by the inductive hypothesis.

If $i \geq |G(\bar{x}, \bar{X})|$ then we check all bits of $H(i - |G(\bar{x}, \bar{X})|, \bar{x}, \bar{X})$ and take AND of them. This can be computed by a single application of unbounded fan-in AND of algorithm for H. Again, we conclude that it is checked in $\mathcal{C}$ by the inductive hypothesis and the closure of $\mathcal{C}$ under AC^0 operations. □

2.2 Two-sort bounded arithmetic

We consider two types of bounded arithmetic theories which turns out to be equivalent.

The first type is based on the Lindstöm quantifier. First we briefly summarize the concept of the Lindström quantifier.

Let $\sigma = \langle P_1, \cdots, P_s \rangle$ be a signature where $P_1, \ldots, P_s$ are relation symbols and $\mathcal{K} \subseteq Struct(\sigma)$ be a set which is complete for a given complexity class $\mathcal{C}$.

Let $\tau = \langle R_1, \ldots, R_k, c_1, \ldots, c_l \rangle$ be a signature where $R_1, \ldots, R_k$ are relation symbols and $c_1, \ldots, c_l$ are constant symbols. For τ formulae $\phi_1, \ldots, \phi_s$ we define the Lindström quantifier $Q_{\mathcal{K}}$ as

$$\mathcal{A} \models Q_{\mathcal{K}}[\phi_1, \ldots, \phi_s] \Leftrightarrow (univ(\mathcal{A}), \phi_1^{\mathcal{A}}, \ldots, \phi_s^{\mathcal{A}}) \in \mathcal{K}.$$

for any $\mathcal{A} \in Struct(\tau)$.

When we consider the subclass $\mathcal{C}$ of P, the membership relation

$$(univ(\mathcal{A}), \phi_1^{\mathcal{A}}, \ldots, \phi_s^{\mathcal{A}}) \in \mathcal{K}.$$

can be described by some Σ_1^B relation. We will call such relation as the Σ_1^B description of $\mathcal{K}$, that is,

Definition 3 *The set $\mathcal{K} \subseteq Struct(\sigma)$ has a Σ_1^B description $\varphi(n, P_1, \ldots, P_s) \in \Sigma_1^B$ if for all n, $P_1, \ldots, P_s$,*

$$\varphi(n, P_1, \ldots, P_s) \Leftrightarrow (\{0, \ldots, n-1\}, P_1, \ldots, P_s) \in \mathcal{K}$$

holds in the standard model.

Now we shall give a presentation of Lindström quantifier in two sort systems. The idea is to give a description of the satisfaction relation $\mathcal{A} \models Q_{\mathcal{K}}[\phi_1, \ldots, \phi_s]$ in the language L_2. Note that a τ-structure is coded by a tuple

$$\langle n, c_1, \ldots, c_l, R_1, \ldots, R_k \rangle$$

where n is the size of the universe. The description of the above satisfaction relation is obtained by replacing $P_1, \ldots, P_s$ by $\phi_1^{\mathcal{A}}, \ldots, \phi_s^{\mathcal{A}}$ respectively.

Definition 4 *Let $\varphi_{\mathcal{K}}(n, P_1, \ldots, P_s)$ be a Σ_1^B description of $\mathcal{K}$. Let $\phi_1, \ldots, \phi_s$ be L_2 formulae. Then we define $\varphi_{\mathcal{K}}(n, \bar{c}, \bar{R}, \phi_1, \ldots, \phi_s)$ as the L_2 formula which is obtained from $\varphi_{\mathcal{K}}$ by replacing all occurrences of $P_i(x_1, \ldots, x_{t_i})$ by $\phi_i(x_1, \ldots, x_{t_i}, c_1, \ldots, c_l, R_1, \ldots, R_k)$ for $1 \leq i \leq s$. We call this scheme the Σ_1^B description of $Q_{\mathcal{K}}$ over the signature τ.*

Let Φ be a class of L_2 formulae. We define $\varphi_{\mathcal{K}}(\Phi)$ to be the class of formulae of the form $\varphi_{\mathcal{K}}(n, \bar{c}, \bar{R}, \phi_1, \ldots, \phi_s)$ where $\phi_1, \ldots, \phi_s \in \Phi$.

Based on this argument we define a L_2-system as follows:

Definition 5 *Let $\mathcal{K} \subset Struct(\sigma)$ and $\varphi_{\mathcal{K}}$ be the Σ_i^B description of $Q_{\mathcal{K}}$ over the signature $\tau = \langle i, c_1, \ldots, c_l, R_1, \ldots, R_k \rangle$. The L_2 theory $\mathbf{V}$-$Q_{\mathcal{K}}$ consists of the following axioms:*

- *BASIC*
- *$\varphi_{\mathcal{K}}(\Sigma_0^B)$-COMP*

In the next section we will give a general theory for when the system $\mathbf{V}$-$Q_{\mathcal{K}}$ captures the corresponding complexity class.

3 Witnessing theorem via complete problems

Our formulation of $\mathbf{V}$-$Q_{\mathcal{K}}$ resembles to the theory defined by Kolokolova [3]. So we modify Kolokolova's criteria for $\mathbf{V}$-Φ to capture a complexity class in order to suit with our formulation.

Nevertheless, our proof uses slightly different approach, namely, we will define another theory $\mathbf{V}$-$\mathcal{K}$ based on the complete problem $\mathcal{K}$ which affords a direct application of Herbrand's theorem and show the equivalence with $\mathbf{V}$-$Q_{\mathcal{K}}$.

First let us see how $\mathbf{V}$-$\mathcal{K}$ is defined. Let $\mathcal{K}$ be a complete problem as above and let $\varphi_{\mathcal{K}}$ be its Σ_1^B description. Suppose that the complement $\mathcal{K}^c$ of $\mathcal{K}$ also has a Σ_1^B description, say $\psi_{\mathcal{K}}$. Note that this condition follows from another condition that the class $\mathcal{C}$ is closed under complementation.

We define the theory $\mathbf{V}$-$\mathcal{K}$ as follows:

Definition 6 *The L_2-theory $\mathbf{V}$-$\mathcal{K}$ has the following axioms:*

- *BASIC*
- Σ_0^B*-COMP*
- *the existence of witnesses for $\mathcal{K} \cup \mathcal{K}^c$:*

$$(\forall n)(\forall P_1)\cdots(\forall P_s)\,(\varphi_{\mathcal{K}}(n, P_1, \ldots, P_s) \vee \psi_{\mathcal{K}}(n, P_1, \ldots, P_s))$$

We say that $\mathbf{V}$-$\mathcal{K}$ is *properly defined* if there exist Σ_1^B descriptions $\varphi_{\mathcal{K}}$ and $\psi_{\mathcal{K}}$ for $\mathcal{K}$ and $\mathcal{K}^c$ respectively.

We give a slight modification of Kolokolova's criteria for a system to capture a complexity class as follows:

- **Strong closure** : $\varphi_{\mathcal{K}}(\Sigma_0^B)$ is *strongly closed* if for all $\psi \in \Sigma_0^B(\varphi_{\mathcal{K}}(\Sigma_0^B))$ there exists $\eta \in \varphi_{\mathcal{K}}(\Sigma_0^B)$ such that
$$\mathbf{V}\text{-}Q_{\mathcal{K}} \vdash \psi \leftrightarrow \eta.$$
- **Self-witnessing** : $\varphi_{\mathcal{K}}(\Sigma_0^B)$ is *self-witnessing* if for all Σ_1^B description $(\exists Z < t)\varphi_{\mathcal{K}}^0(\bar{x}, \bar{X}, Z)$ if $\mathbf{V}$-$Q_{\mathcal{K}} \vdash (\forall \bar{x})(\forall \bar{X})(\exists Z < t)\varphi_{\mathcal{K}}^0(\bar{x}, \bar{X}, Z)$ then there exists a function $F(\bar{x}, \bar{X})$ which is bitwise computable in $\mathbf{V}$-$Q_{\mathcal{K}}$ such that
$$\mathbf{V}\text{-}Q_{\mathcal{K}} \vdash (\forall \bar{x})(\forall \bar{X})\varphi_{\mathcal{K}}^0(\bar{x}, \bar{X}, F(\bar{x}, \bar{X})).$$

Now we state and prove our main theorems.

Theorem 2 *Let $\varphi_{\mathcal{K}}(\Sigma_0^B)$ be a logic which captures the complexity class $\mathcal{C}$ over a given signature τ. Suppose that $\varphi_{\mathcal{K}}(\Sigma_0^B)$ is strongly closed and constructive. If $\mathbf{V}$-$Q_{\mathcal{K}}$ contains V^0 then $\mathbf{V}$-$Q_{\mathcal{K}}$ is equivalent to $\mathbf{V}$-$\mathcal{K}$.*

We will use the conservative universal extension of $\mathbf{V}$-$\mathcal{K}$ to prove theorem 2. So we first define this and prove its conservation over $\mathbf{V}$-$\mathcal{K}$.

Intuitively, the universal extension $\overline{\mathbf{V}\text{-}\mathcal{K}}$ is obtained by introducing Skolem functions to eliminate existential quantifiers.

Let pd, f_{SE} and $F_{\mathcal{K}}$ be new function symbols. We define the language $L_{\mathcal{K}}$ to be the smallest class satisfying:

- The function symbols pd, f_{SE} and $F_{\mathcal{K}}$ are in $L_{\mathcal{K}}$.
- For each open $L_{\mathcal{K}}$ formula $\varphi(z, \bar{x}, \bar{X})$ and an L_2 term $t(\bar{x}, \bar{X})$, there are a string function $F_{\varphi,t}$ and a number function $F_{\varphi,t}$ in $L_{\mathcal{K}}$.

The system $\overline{\mathbf{V}\text{-}\mathcal{K}}$ consists of axioms for all symbols in $L_{\mathcal{K}}$.

Definition 7 *The $L_{\mathcal{K}}$ system $\overline{\mathbf{V}\text{-}\mathcal{K}}$ consists of the following axioms:*

- *BASIC*
- $pd(0) = 0$, $pd(x) + 1 = x$,
- *Extensionality:*

$$(f_{SE}(X,Y) \leq |X|) \wedge (z < f_{SE}(X,Y) \rightarrow (X(z) \leftrightarrow Y(z)))$$
$$\wedge (f_{SE}(X,Y) < |X| \rightarrow (X(f_{SE}(X,Y)) \not\leftrightarrow Y(f_{SE}(X,Y)))).$$

- *Witnessing $\mathcal{K} \cup \mathcal{K}^c$:*

$$(\forall n)(\forall P_1)\cdots(\forall P_s)(\varphi^0_{\mathcal{K}}(n, P_1, \ldots, P_s, F_{\mathcal{K}}(n, P_1, \ldots, P_s))$$
$$\vee \psi^0_{\mathcal{K}}(n, P_1, \ldots, P_s, F_{\mathcal{K}}(n, P_1, \ldots, P_s)))$$

- *axiom for $F_{\varphi,t}$:*

$$z \in F_{\varphi,t}(\bar{x}, \bar{X}) \leftrightarrow (z < t(\bar{x}, \bar{X}) \wedge \varphi(z, \bar{x}, \bar{X}))$$

- *axiom for $f_{\varphi,t}$:*

$$f_{\varphi,t}(\bar{x}, \bar{X}) \leq t \wedge (w < f_{\varphi,t}(\bar{x}, \bar{X}) \rightarrow \neg\varphi(w, \bar{x}, \bar{X}))$$
$$\wedge (f_{\varphi,t}(\bar{x}, \bar{X}) < t \rightarrow \varphi(f_{\varphi,t}(\bar{x}, \bar{X}), \bar{x}, \bar{X})).$$

Proposition 1 *For any $\varphi \in \Sigma^B_0(L_{\mathcal{K}})$ there exists an open $L_{\mathcal{K}}$ formula φ' such that $\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash \varphi \leftrightarrow \varphi'$.*

(Proof) By induction on the complexity of $\varphi \in \Sigma^B_0$. We prove for the case where the outermost connective is a bounded number quantifier. Let $\varphi \equiv (\exists x < t)\varphi_0(x)$. By the inductive hypothesis, we have an open $L_{\mathcal{K}}$ formula φ'_0 such that $\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash \varphi_0 \leftrightarrow \varphi'_0$. So we have

$$\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash (\exists x < t)(\varphi_0(x) \leftrightarrow \varphi'_0(f_{\varphi,t}(x))).$$

The universal case can be treated as $(\forall x < t)\varphi_0(x) \equiv \neg(\exists x < t)\neg\varphi_0(x)$. □

Theorem 3 *Assume that $\mathbf{V}\text{-}\mathcal{K}$ is properly defined. Then $\overline{\mathbf{V}\text{-}\mathcal{K}}$ is a conservative extension of $\mathbf{V}\text{-}\mathcal{K}$.*

(Proof). First we show that $\overline{\mathbf{V}\text{-}\mathcal{K}}$ extends $\mathbf{V}\text{-}\mathcal{K}$. It suffices to show that $\overline{\mathbf{V}\text{-}\mathcal{K}}$ proves Σ^B_0-COMP. Let $\varphi \in \Sigma^B_0$. By Proposition 1 there exists φ' such that $\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash \varphi \leftrightarrow \varphi'$. By the axiom for $F_{\varphi',t}$ we have

$$(\forall z < t)(z \in F_{\varphi',t}(\bar{x}, \bar{X}) \leftrightarrow \varphi'(z, \bar{x}, \bar{X}))$$

which immediately implies COMP axiom for $\varphi' \equiv \varphi$.

Next we prove the conservation. Above argument shows that $\overline{\mathbf{V}\text{-}\mathcal{K}}$ contains $\overline{\mathbf{V}^0}$. So we take $\overline{\mathbf{V}^0}$ as the base theory and define an infinite chain of theories.

Let $\mathcal{L}_0$ be the language of $\overline{\mathbf{V}^0}$, $\mathcal{L}_1 = \mathcal{L}_0 \cup \{F_{\mathcal{K}}\}$ and

$$\mathcal{L}_{n+1} = \mathcal{L}_n \cup \{F_{\varphi,t} : \varphi \text{ is an open } \mathcal{L}_n \text{ formula}\}$$
$$\cup \{f_{\varphi,t} : \varphi \text{ is an open } \mathcal{L}_n \text{ formula}\}$$

Define $\overline{\mathbf{V}\text{-}\mathcal{K}}_n$ to be the $\mathcal{L}_n$-theory whose axioms are those of $\overline{\mathbf{V}\text{-}\mathcal{K}}$ restricted to $\mathcal{L}_n$. Thus $\overline{\mathbf{V}\text{-}\mathcal{K}} = \bigcup_{n\in\omega} \overline{\mathbf{V}\text{-}\mathcal{K}}_n$.

We will show that $\overline{\mathbf{V}\text{-}\mathcal{K}}_{n+1}$ is conservative over $\overline{\mathbf{V}\text{-}\mathcal{K}}_n$. This is proved by showing that each $F_{\varphi,t}$ and $f_{\varphi,t}$ in $\mathcal{L}_{n+1}$ is definable in $\overline{\mathbf{V}\text{-}\mathcal{K}}_n$. Since $F_{\varphi,t}$ and $f_{\varphi,t}$ are defined by COMP axiom and minimization principle for $\varphi \in \Sigma_0^B(\mathcal{L}_n)$ respectively, it suffices to show that

$$\overline{\mathbf{V}\text{-}\mathcal{K}}_n \vdash \Sigma_0^B(\mathcal{L}_n)\text{-}COMP \text{ and } \overline{\mathbf{V}\text{-}\mathcal{K}}_n \vdash \Sigma_0^B(\mathcal{L}_n)\text{-}MIN.$$

The argument which follows is completely analogous to Nguyen's proof of Theorem 3.11 a) in [5] so we omit it. □

(Proof of Theorem 2) First we prove that $\mathbf{V}\text{-}Q_\mathcal{K}$ contains $\mathbf{V}\text{-}\mathcal{K}$. By assumption, it suffices to show that $\mathbf{V}\text{-}Q_\mathcal{K}$ proves the axiom

$$(\forall n)(\forall P_1)\cdots(\forall P_s)\,(\varphi_\mathcal{K}(n, P_1, \ldots, P_s) \vee \psi_\mathcal{K}(n, P_1, \ldots, P_s))\,.$$

By assumption, $\mathbf{V}\text{-}Q_\mathcal{K}$ proves the closure of $\varphi_\mathcal{K}(\Sigma_0^B)$ under complementation. So in particular, it proves

$$(\forall n)(\forall P_1)\cdots(\forall P_s)\,(\varphi_\mathcal{K}(n, P_1, \ldots, P_s) \leftrightarrow \neg\psi_\mathcal{K}(n, P_1, \ldots, P_s))\,.$$

Therefore we have $\neg\psi_\mathcal{K} \rightarrow \varphi_\mathcal{K}$ which is equivalent to $\psi_\mathcal{K} \vee \varphi_\mathcal{K}$.

For the converse inclusion, we show that $\mathbf{V}\text{-}\mathcal{K}$ proves the COMP axiom for $\varphi_\mathcal{K}(\phi_1, \ldots, \phi_s 0)$ for all $\phi_1, \ldots, \phi_s \in \Sigma_0^B$. We reason in $\overline{\mathbf{V}\text{-}\mathcal{K}}$. First we use Σ_0^B-COMP axiom to convert formulae $\phi_1, \ldots, \phi_s$ into strings $P_1, \ldots, P_s$ such that

$$(\forall x < t_i)(x \in P_i \leftrightarrow \phi_i(x))$$

for $1 \leq i \leq s$. By the Skolemized version of the axiom

$$(\forall \bar{X})(\varphi_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s) \vee \psi_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s))$$

we have

$$(\forall \bar{X})(\varphi^0_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s, F_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s)) \\ \vee \psi^0_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s, F_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s))).$$

Note that $\varphi_\mathcal{K}, \psi_\mathcal{K} \in \Sigma_0^B$. So we can apply COMP axiom to

$$\varphi^0_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s, F_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s))$$

to obtain a string $Y < a$ such that

$$(\forall x < a)(x \in Y \leftrightarrow \varphi^0_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s, F_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s))).$$

Thus

$$(\exists Y < a)(\forall x < a)(x \in Y \leftrightarrow \varphi_\mathcal{K}(x, \bar{X}, P_1, \ldots, P_s))$$

is provable in $\overline{\mathbf{V}\text{-}\mathcal{K}}$ and hence by Theorem 3 in $\mathbf{V}\text{-}\mathcal{K}$. This proves our claim. □

Theorem 4 *Let $\varphi_\mathcal{K}(\Sigma_0^B)$ be a logic which capture the complexity class $\mathcal{C}$ over a given signature τ. If $\varphi_\mathcal{K}(\Sigma_0^B)$ is strongly closed and constructive then $\mathbf{V}\text{-}\mathcal{K}$ captures $\mathcal{C}$.*

First we prove a technical lemma.

Lemma 1 *A function is Σ_1^B definable in $\overline{\mathbf{V}\text{-}\mathcal{K}}$ if and only if it is in $L_\mathcal{K}$.*

(Proof). First note that $\overline{\mathbf{V}\text{-}\mathcal{K}}$ is an universally axiomatized theory. Let $\varphi \in \Sigma_1^B$. Without loss of generality, we can assume that φ is of the form

$$(\exists \bar{Z} < \bar{t})\varphi_0(\bar{x}, \bar{X}, Y, \bar{Z}).$$

Suppose that $\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash (\forall \bar{x})(\forall \bar{X})(\exists Y)(\exists \bar{Z})\varphi_0(\bar{x}, \bar{X}, Y, \bar{Z})$. Then by Herbrand's theorem, there exist $L_{\mathcal{K}}$-terms $F(\bar{x}, \bar{X})$, $\bar{G}(\bar{x}, \bar{X})$ such that

$$\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash (\forall \bar{x})(\forall \bar{X})\varphi_0(\bar{x}, \bar{X}, F(\bar{x}, \bar{X}), \bar{G}(\bar{x}, \bar{X})).$$

We claim that $L_{\mathcal{K}}$ is closed under complementation.

For simplicity, let $F(\bar{x}, \bar{X}) = G(H(\bar{x}, \bar{X}))$ where

$$(\forall y < t(X))(y \in G(X) \leftrightarrow \varphi(y, X)),$$
$$(\forall y < s(\bar{x}, \bar{X}))(y \in H(\bar{x}, \bar{X}) \leftrightarrow \psi(\bar{x}, y, \bar{X}))$$

where φ, ψ are open $L_{\mathcal{K}}$ formulae and t, s are L_2 terms. Let $\Phi(\bar{x}, y, \bar{X}) \equiv \varphi(y, H(\bar{x}, \bar{X})$ and $r(\bar{x}, \bar{X}) = t(s(\bar{x}, \bar{X}))$. Then it is easy to see that

$$(\forall y < r(\bar{x}, \bar{X}))(y \in F(\bar{x}, \bar{X}) \leftrightarrow \Phi(\bar{x}, y, \bar{X})).$$

Thus we have proved the only-if part.

The if-part is immediately implied by defining axioms of functions in $L_{\mathcal{K}}$. □

(Proof of Theorem 4). By Theorem 3, it suffices to show that $\overline{\mathbf{V}\text{-}\mathcal{K}}$ captures $\mathcal{C}$. Let $\varphi \in \Sigma_1^B$ and suppose that

$$\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash (\forall \bar{x})(\forall \bar{X})(\exists Y)\varphi(\bar{x}, \bar{X}, Y).$$

By Σ_0^B-REPL in $\overline{\mathbf{V}\text{-}\mathcal{K}}$ we may assume that φ is in strict form. So we have

$$\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash (\forall \bar{x})(\forall \bar{X})(\exists Y)(\exists \bar{W})\varphi_0(\bar{x}, \bar{X}, Y, \bar{W})$$

for $\varphi_0 \in \Sigma_0^B$. Since $\overline{\mathbf{V}\text{-}\mathcal{K}}$ is an universal axiomatized theory, we can apply Herbrand's theorem. So there exist $L_{\mathcal{K}}$ terms $F, \bar{G}$ such that

$$\overline{\mathbf{V}\text{-}\mathcal{K}} \vdash (\forall \bar{x})(\forall \bar{X})\varphi_0(\bar{x}, \bar{X}, F(\bar{x}, \bar{X}), \bar{G}(\bar{x}, \bar{X})).$$

Thus it suffices to show that the functions in $L_{\mathcal{K}}$ coincide with those bitwise computable in $\mathcal{C}$.

First we show that any function in $L_{\mathcal{K}}$ is bitwise computable in $\mathcal{C}$. The functions pd and f_{SE} are known to be AC^0 computable. For the function $F_{\mathcal{K}}$, we use the constructiveness property of $\varphi_{\mathcal{K}}(\Sigma_0^B)$. For functions of the form $F_{\varphi,t}$ and $f_{\varphi,t}$ we show that for all $n \geq 1$, $F_{\varphi,t}$ and $f_{\varphi,t}$ in $\mathcal{L}_n$ is in $\mathcal{FC}$ by induction on n.

For $n = 1$ this is trivial. Let $F_{\varphi,t} \in \mathcal{L}_{n+1}$. Then for an open $\mathcal{L}_n$ formula φ and an L_2-term t

$$(\forall z < t)(z \in F_{\varphi,t}(\bar{x}, \bar{X} \leftrightarrow \varphi(z, \bar{x}, \bar{X})).$$

This function $F_{\varphi,t}$ is defined by the concatenation recursion on notation. First define

$$G(n, \bar{x}, \bar{X}) = \begin{cases} \varepsilon & \text{if } n = 0, \\ G(n-1, \bar{x}, \bar{X}) * 0 & \text{if } \varphi(n, \bar{x}, \bar{X}), \\ G(n-1, \bar{x}, \bar{X}) * 1 & \text{if } \neg\varphi(n, \bar{x}, \bar{X}). \end{cases}$$

Then we have $F_{\varphi,t}(\bar{x}, \bar{X}) = G(t, \bar{x}, \bar{X})$. Since $\mathcal{FC}$ is closed under the CRN operation, $F_{\varphi,t}$ is in $\mathcal{FC}$.

Let $f_{\varphi,t} \in \mathcal{L}_{n+1}$. Then

$$f_{\varphi,t}(\bar{x}, \bar{X}) = \mu z < t\varphi(z, \bar{x}, \bar{X}).$$

It is known (Clote-Takeuti) that AC^0 is closed under sharply bounded minimization. So $f_{\varphi,t}$ is in $\mathcal{FC}$.

Conversely, let F or f be in $\mathcal{FC}$. Recall that $\mathcal{FC}$ is the AC^0 closure of the function $F_{\mathcal{K}}$ which is bitwise decidable by $\mathcal{K}$.

We prove by induction on the complexity of $F \in \mathcal{FC}$ that F is in $L_{\mathcal{K}}$.

The base functions of AC^0 are trivially in $L_{\mathcal{K}}$. $F_{\mathcal{K}}$ can be defined as $F_{\mathcal{K},t}$. Furthermore, closure under composition is already proved.

Let $F(n, \bar{x}, \bar{X})$ be defined by CRN as

$$\begin{aligned} F(0, \bar{x}, \bar{X}) &= G(\bar{x}, \bar{X}) \\ F(n, \bar{x}, \bar{X}) &= \begin{cases} F(n, \bar{x}, \bar{X}) * 0 & \text{if } \neg\varphi(n, \bar{x}, \bar{X}), \\ F(n, \bar{x}, \bar{X}) * 1 & \text{if } \varphi(n, \bar{x}, \bar{X}). \end{cases} \end{aligned}$$

Define $t(n, \bar{x}, \bar{X}) = |G(\bar{x}, \bar{X})| + n$ and

$$\Phi(z, \bar{x}, \bar{X}) \equiv (z < |G(\bar{x}, \bar{X})| \to z \in G(\bar{x}, \bar{X})) \wedge (z \geq |G(\bar{x}, \bar{X})| \to \varphi(z, \bar{x}, \bar{X})).$$

Then we have $F(n, \bar{x}, \bar{X}) = F_{\Phi,t}(\bar{x}, \bar{X})$ which ends the proof. □

Corollary 1 *Let $\varphi_{\mathcal{K}}(\Sigma_0^B)$ be a logic which capture the complexity class $\mathcal{C}$ over a given signature τ. If $\varphi_{\mathcal{K}}(\Sigma_0^B)$ is strongly closed and constructive then $\mathbf{V}$-$Q_{\mathcal{K}}$ captures $\mathcal{C}$.*

4 A theory based on acyclic conjunctive queries

As an application of the witnessing argument in the previous section, we will construct a theory based on the Acyclic Conjunctive Query problem. Let Boolean Conjunctive Query problem be the following problem:

> Given a relational database **db** and a query Q which is a conjunction of relations in **db**, decide whether there exists an assignment of values to attributes appearing in Q so that Q evaluates to true on **db** by the assignment.

Then an Acyclic Boolean Conjunctive Query problem (ABCQ) is a BCQ in which the query Q is restricted to have an acyclic hypergraph representation.

Gottlob et.al. [1] proved the following

Theorem 5 (Gottlob et.al.) *ABCQ is complete for LOGSPACE via AC^0 reductions. Furthermore, it remains complete even if all relations are restricted to binary.*

We will use the binary version of ABCQ to formulate our theory. First we shall formulate the problem over finite structures of some signature. Let $\sigma_{db} = \{D, Q\}$ where D and Q are binary predicates. Intuitively, D and Q represent a database and a query such that

$$D(x, y) \Leftrightarrow \langle x, y \rangle \text{ is a record for the relation in } D$$

and

$$Q(i, j) \Leftrightarrow \langle i, j \rangle \text{ is a conjunct in } Q.$$

Thus we will define a complete set

$$\mathcal{K}_{ABCQ} = \{\langle [n], Q, D \rangle \ : \text{ the query } Q \text{ has a solution in the database } D\}.$$

Let Q_{ABCQ} be the Lindström quantifier for the set $\mathcal{K}_{ABCQ}$. Observing that ABCQ is complete for LOGCFL, it is not difficult to see that

Proposition 2 *The logic $Q_{ABCQ}(FO)$ captures LOGCFL over arbitrary structure.*

Next we define a Σ_1^B description of $\mathcal{K}_{ABCQ}$. A solution S is a mapping $[n] \to [n]$. So the following formula expresses that the query Q is true on D by the solution S;

$$CQ(D,Q,S) \Leftrightarrow (\forall i)(\forall j)(\forall x)(\forall y)((Q(i,j) \to D(S[i]S[j])).$$

Thus

$$(\exists S)CQ(D,Q,S) \Leftrightarrow \text{the query } Q \text{ is true on } D.$$

The property of $\mathcal{K}_{ABCQ}$ is expressed as: either Q is cyclic or Q is true on D. So we define

$$Cyclic(Q) \Leftrightarrow (\exists C)(\forall i)(Q(C[i],C[i+1]) \wedge (\exists i)(\exists j)(C[i] = C[j])).$$

Finally let

$$\varphi_{ABCQ}(n,Q,D) \Leftrightarrow Cyclic(Q) \vee (\exists S)CQ(D,Q,S).$$

Let $\varphi_D(x,y,n,\bar{c},\bar{R})$ and $\varphi_Q(i,j,n,\bar{c},\bar{R})$ be L_2 formulae. By replacing occurrences of D and Q by φ_D and φ_Q respectively, we obtain the Σ_1^B description of $\mathcal{K}_{ABCQ}$ in the language L_2 as follows:

$$\begin{array}{l}\varphi_{ABCQ}(n,\bar{c},\bar{R},\varphi_D,\varphi_Q) \Leftrightarrow \\ (\exists C)(\forall i)(\varphi_Q(C[i],C[i+1],n,\bar{c},\bar{R}) \wedge (\exists i)(\exists j)(C[i] = C[j]))\vee \\ (\exists S)(\forall i)(\forall j)(\forall x)(\forall y)((\varphi_Q(i,j,n,\bar{c},\bar{R}) \to \varphi_D(S[i],S[j],n,\bar{c},\bar{R})).\end{array}$$

Definition 8 *The L_2-theory $\mathbf{V}$-Q_{ABCQ} consists of the following axioms:*

- *BASIC*
- *$\varphi_{ABCQ}(\Sigma_0^B)$-COMP*

Using the framework given in the previous section, we will show that $\mathbf{V}$-Q_{ABCQ} captures LOGCFL. To this end, we first need some technical tools.

To prove strong closure and self-witnessing of $\mathbf{V}$-Q_{ABCQ}, we use the circuit characterization of LOGCFL, that is SAC^1. In particular, we will make use of a particular form of SAC^1 circuits. A SAC^1 circuit is in normal form if it satisfies the followings:

- there is a single output gate.
- all internal gates g receives inputs from gates on depth $depth(g) - 1$.
- all gates on even depth are AND gates, while all gates on odd depth > 1 are OR gates,
- gates on depth 1 are input gates all of which are either x_i, $\neg x_i$ or constant true.

We define a formula expressing that a SAC^1 circuit in normal form with depth d, width w and ψ as its DCL accepts input X of length n. First define a Σ_0^B formula $NF_\psi(n)$ which says that ψ is a DCL of an SAC^1 circuit in normal form with input of length n. Next we define a formula expressing the conditions for proof trees. Notice that the topological structure of the witness for SAC^1 circuits in normal form is uniquely determined by the depth as $SKEL_d$. So we code it as

$$SKEL_d = \{\langle i,j\rangle : 0 \le i \le d, 0 \le j < \lceil i/2 \rceil\}.$$

Now we define

$$\begin{array}{l}ProofTree(n,d,w,T,X) \Leftrightarrow \\ T : SKEL_d \to [d \cdot w]\wedge \\ (\forall i < d)(\forall j < 2\lceil i/2 \rceil)(depth(T(i,j)) = i \wedge width(T(i,j)) < w)\wedge \\ (Even(i) \to (\psi(T(i,j),T(i+1,2j),n) \wedge \psi(T(i,j),T(i+1,2j+1),n)))\wedge \\ (Odd(i) \to (\psi(T(i,j),T(i+1,j),n)))\wedge \\ (\forall j < 2\lceil i/2 \rceil)((l(T(0,j)) = x_i \wedge X(i)) \vee (l(T(0,j)) = \neg x_i \wedge \neg X(i))).\end{array}$$

Finally let

$$\varphi_{NF}[\psi](n,d,w,X) \Leftrightarrow NF_{\psi}(n) \rightarrow (\exists T)ProofTree(n,d,w,T,X).$$

Based on these formulations we have

Theorem 6 *for any $\psi \in \Sigma_0^B$ there exist $\eta_D, \eta_Q \in \Sigma_0^B$ such that* **V**-Q_{ABCQ} *proves*

$$(\forall X)(\exists \bar{c})(\exists \bar{R})(\varphi_{NF}(|X|,d,w,X) \leftrightarrow \varphi_{ABCQ}(|X|,\bar{c},\bar{R},\eta_D,\eta_Q)).$$

(Proof). Let ψ be a DCL of an SAC^1 circuit family in normal form. We will define the descriptions of a database η_D and a query η_Q for ψ as in the proof of Gottlob.

Firstly, η_Q is given by coding $SKEL_d$, thus can be determined solely by the depth d as

$$\begin{aligned}\eta_Q(i,j) \quad \Leftrightarrow \quad & Pair(i) \wedge Pair(j) \wedge (i)_0 = (j)_0 + 1 \wedge \\ & \wedge (Even((i)_0) \rightarrow ((j)_1 = 2(i)_1 \vee (j)_1 = 2(i)_1 + 1)) \\ & \wedge (Odd(i)_0 0 \rightarrow (j)_1 = (i)_1).\end{aligned}$$

Secondly, η_D is the database expression of the circuit defined by η with the input bits and negations that are labeled by 1 at the input level.

$$\begin{aligned}\eta_D(x,y,n,\bar{c},\bar{R}) \quad \Leftrightarrow \quad & \psi(x,y,n) \wedge \\ & (depth(x) = 1 \rightarrow (l(x) = 1 \leftrightarrow width(x) \in bin(\bar{c},\bar{R}))).\end{aligned}$$

Let R be an unary predicate such that $R(i) \Leftrightarrow i \in X$ and consider the signature $\langle R \rangle$. Let S be a solution to the ABCQ instance defined by η_D and η_Q. Then S directly gives the mapping $SKEL_d \rightarrow [d \cdot w]$ satisfying $ProofTree(n,d,w,T,X)$. □

As a corollary, we have

Corollary 2 **V**-Q_{ABCQ} *proves* Σ_0^B-*COMP.*

(Proof). The idea is to convert a given Σ_0^B formula φ into an SAC^1 circuit family and apply Theorem 6. Since Σ_0^B formulae are converted into an AC^0 circuit family in a natural manner, this can be readily done. □

Theorem 7 $\varphi_{ABCQ}(\Sigma_0^B)$ *is strongly closed.*

The hardest part of the proof of Theorem 7 is to show the closure under complementation. The idea is to define computations of SAC^1 circuits inside **V**-Q_{ABCQ} and formalize the proof the closure of SAC^1 under complementation.

We will omit the details here since it is long and tedious.

4.1 Self witnessing property for V-Q_{ABCQ}

Next we prove the self witnessing property of **V**-Q_{ABCQ}. The idea is to formalize the proof of the following theorems:

Theorem 8 (Gottlob et.al.) *Let M be a bounded tree-size logspace ATM recognizing A. It is possible to construct a L^{LOGCFL} transducer which for input $w \in A$ outputs a single accepting tree for M on w.*

Theorem 9 (Gottlob et.al.) *Computing a solution to an acyclic CSP instance (if any) is feasible in L^{LOGCFL}.*

We will first overview the outline of our proof. Let $(\exists Z < t)\varphi_0(\bar{x}, \bar{X}, Z)$ be a Σ_1^B description of $\varphi_{ABCQ}(\Sigma_0^B)$. Describe an ATM algorithm deciding it, as in Theorem 8. It can be shown that the formalization of of ATM algorithms can be done in $\mathbf{V}^0$.

Secondly, formalize the proof of Theorem 8 and Theorem 9 in $\mathbf{V}$-$Q_{\mathcal{K}}$, the essential change is that we use bitwise $\varphi_{ABCQ}(\Sigma_0^B)$ computable functions instead of L^{LOGCFL} transducer.

Finally extract a witness Z for $\varphi_0(\bar{x}, \bar{X}, Z)$ from the accepting tree of the ATM given as above.

First we will show that computations of bounded tree-size logspace ATMs can be coded inside $\mathbf{V}$-Q_{ABCQ}.

Definition 9 *A (two tape) alternating Turing machine (ATM) is a tuple* $M = (\Sigma, Q, q_0, \delta, g)$ *where* Σ *is an alphabet,* Q *is the set of states,* q_0 *is the initial state,* $\delta : Q \times \Sigma^2 \to P(Q \times \Sigma \times \{-1, 0, 1\}^2)$ *is the transition function, and* $g : Q \to \{\wedge, \vee, 0, 1\}$ *is the state function.*

The intended meaning of δ is that if $(q', j', m, n) \in \delta(q, i, j)$ then M can make the following move:

- change the state from q to q'
- rewite the letter of work tape head from j to j'
- move heads of input and work tapes to m and n respectively,

An accepting computation tree of an ATM M on input X is a labeled rooted tree such that

- the root is labeled by the initial configuration of M on input X,
- if a node is labeled by a configuration c with a state q such that $g(q) = \wedge$ then it has a offspring for each configuration which can be moved from c in one step according to the transition function,
- if a node is labeled by a configuration c with a state q such that $g(q) = \vee$ then it has a single offspring which can be moved from c in one step according to the transition function,
- each leaf node is labeled by a configuration with a state q such that $g(q) = 1$.

An ATM M accepts an input X if there exists an accepting computation tree of M on input X. The tree-size of an accepting tree of M on an input X is the number of nodes in it. The space of an accepting tree is the maximal number of cells used in the computation.

Definition 10 *For functions* $s(n)$ *and* $t(n)$*, we define* $ASPACE-TREESIZE(s(n), t(n))$ *to be the class of predicates which are decidable by* $s(n)$ *tree-size,* $t(n)$ *space bounded ATMs.*

We will express computations of ATMs by the formula which asserts the existence of accepting trees. An accepting tree is coded by a two dimensional array T such that

- Each column of T corresponds to a node in the tree,
- $T[0]$ is the root node corresponding to the initial configuration,
- For all $i > 0$ $T[i]$ has an unique direct ancestor $T[j]$ where $j < i$,
- if $T[i]$ and $T[i']$ have direct ancestors $T[j]$ and $T[j']$ respectively and $i < i'$ then $j < j'$,
- if $T[i]$ is a node corresponding to an existential configuration then it has a single offspring which is obtained by a single step of M,

- if $T[i]$ is a node corresponding to an universal configuration then it has an offspring for each configuration which is obtained by a single step of M,
- all leaf nodes correspond to accepting configurations.

In order to code configurations of logspace bounded ATMs, we need some coding number functions. We define the pairing function

$$\langle x, y\rangle = \frac{(x+y)(x+y+1)}{2}$$

and for any constant k we define

$$\langle x_1, \ldots, x_k\rangle = \langle x_1, \langle x_2, \ldots, x_k\rangle\rangle.$$

We also define the following predicate.

$$Pair_k(x) \Leftrightarrow (\exists x_1)\cdots(\exists x_k)(x = \langle x_1, \ldots, x_k\rangle).$$

For such pairs we define functions to extract elements from them:

$$el_k^i(x) = \begin{cases} x_i & \text{if } x = \langle x_1, \ldots, x_k\rangle, \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that these functions are Σ_0^B definable in $\mathbf{V}^0$ so we can use them freely in $\mathbf{V}$-Q_{ABCQ}.

Using these functions, we will first give the coding of a configuration of an ATM as

$$\langle q, k, l, w\rangle$$

where $q \in Q$ is the state, k, l are head positions of input and work tapes respectively, and w is the content of the work tape. So we let each node of the accepting tree T have the form

$$T[i] = \langle j, q, k, l, w\rangle$$

where j is the position of its direct ancestor and q, k, l, w as above.

Now we will give an L_2-formula describing accepting trees of ATMs. The first formula says that given T codes a well-formed tree:

$$\begin{aligned} WF(l, s, T) \Leftrightarrow\ & T < l \cdot s \wedge (\forall i < l)(T[i] < s \wedge Pair_5(T[i])) \\ & \wedge el_5^1(T[0]) = 0 \wedge (\forall i < l)(i > 0) \rightarrow el_5^1(T[i]) < i) \\ & \wedge (\forall i, j < l)(i < j \rightarrow el_5^1(T[i]) \leq el_5^1(T[j])) \end{aligned}$$

Since states are divided into five types, namely initial, universal, existential, accepting and rejecting, we distinguish them as follows:

$$\begin{aligned} init(q) &\Leftrightarrow q = 0, \\ univ(q) &\Leftrightarrow q > 0 \wedge q \equiv 0(\mathrm{mod}4), \\ ext(q) &\Leftrightarrow q \equiv 1(\mathrm{mod}4), \\ acc(q) &\Leftrightarrow q \equiv 2(\mathrm{mod}4), \\ rej(q) &\Leftrightarrow q \equiv 3(\mathrm{mod}4). \end{aligned}$$

The transition function of an ATM is given by a finite table M such that

$$M(q, q', u, v, v', m, m') \Leftrightarrow (q', v', m, m') \in \delta(q, u, v).$$

So we define a formula giving the transition relation between configurations of a given ATM as follows:

$$
\begin{array}{l}
Trans(c,c',M,X) \Leftrightarrow \\
(\exists q,k,l,w,q',k',l',w')(c=\langle q,k,l,w\rangle \wedge c'=\langle q,k,l,w\rangle \\
\wedge(\exists -1 \leq m,m' \leq 1)(M(q,q',X(k),bit(w,l),bit(w',l'),m,m')\wedge \\
(\forall p<|w|)(p\neq l \rightarrow bit(w,l)=bit(w',l)) \wedge k'=k+m \wedge l'=l+m')).
\end{array}
$$

where $bit(w,i)$ is the ith bit of w in binary. Let

$$conf(\langle x_1,x_2,x_3,x_4,x_5\rangle)=\langle x_2,x_3,x_4,x_5\rangle$$

and define

$$
\begin{array}{l}
VNode(l,s,i,M,X,T) \equiv \\
(init(el_5^1(T[i])) \vee ext(el_5^1(T[i])) \\
\rightarrow (\exists! j<l)(el_5^1(T[j])=i \wedge Trans(conf(T[i]),conf(T[j]),M,X)))\wedge \\
(univ(el_5^1(T[i])) \\
\rightarrow (\forall c<s)(Trans(conf(T[i]),conf(T[j]),M,X) \\
\quad \leftrightarrow (\exists j)(el_5^1(T[j])=i \wedge conf(T[j])=c)))\wedge \\
(acc(el_5^1(T[i])) \leftrightarrow \neg(\exists j<l)(el_5^1(T[j])=i)).
\end{array}
$$

Finally we define

$$ATREE(l,s,M,X,T) \Leftrightarrow WF(l,s,M,X,T) \wedge (\forall i<l)VNode(l,s,i,M,X,T).$$

We will show that for any Σ_0^B definable M, the statement $(\exists T<l\cdot s)ATREE(l,s,M,X,T)$ is $Q_{ABCQ}(\Sigma_0^B)$ definable in $\mathbf{V}$-Q_{ABCQ}. The idea is to formalize the translation of ATMs by SAC^1 circuit.

Theorem 10 *For any $\varphi \in \Sigma_0^B$ and $l,t,s \in L_2$ there exist $\psi \in \Sigma_0^B$ and $d,w \in L_2$ such that $\mathbf{V}$-Q_{ABCQ} proves*

$$
\begin{array}{l}
(\forall y<t)(M(x) \leftrightarrow \varphi(y)) \rightarrow \\
(\forall X)((\exists T)ATREE(l,s,M,X,T) \leftrightarrow (\exists T')ProofTree_\psi(|X|,d,w,T',X)).
\end{array}
$$

(Proof). We will formalize the direct translation of ATMs into SAC^1 circuits given by Vollmer [7]. We argue inside $\mathbf{V}$-Q_{ABCQ}.

Let M give the transition function of a given ATM with treesize l and space $|s|$. It is easy to see that in $\mathbf{V}$-Q_{ABCQ} we can assume that every configuration of M has at most two successors. For an input X of M we denote the whole computation tree of M on X by $T_M(X)$. Since $T_M(X)$ may be superpolynomial, we do not have a string representing it in general. However, we may have its Σ_0^B graph as

$$CTREE(u,v,M,X) \Leftrightarrow u \text{ is an offspring of } v \text{ in } T_M(X).$$

Note that we can suitably code configurations of M on input X by number objects. We can also code constant numbers of such configurations by numbers. We call a pair $\langle r,s\rangle$ a fragment where r is a configuration and s is a constant number of configurations.

Now we are ready to give formalized proofs of theorems by Gottlob et.al. Firstly, we consider Theorem 8.

The essential part of the proof is to compute the predicate $OCCURS(l,s,c,M,X)$ defined as follows:

$$OCCURS(l,s,c,M,X) \Leftrightarrow (\exists T)(ATREE(l,s,M,X,T) \wedge (\exists i<l)(\exists j<l)(T[i]=\langle j,c\rangle)).$$

Intuitively, it asserts that a given configuration c appears in some accepting computation tree of an ATM M on input X. In [2], it is proven that this predicate can be checked in $LOGCFL$. This statement can be formalized in the following manner.

Lemma 2 $\mathbf{V}$-Q_{SAC} *proves that there exists a* $Q_{SAC}(\Sigma_0^B)$ *formula which is equivalent to the predicate* $OCCURS(l, s, c, M, X)$.

(Proof). The idea of the original proof is to construct an ATM M' which simulates M and in addition checks whether c occurs in some accepting computation tree T. It suffices to show that such ATM M' can be defined by a Σ_0^B formula and thus we can use M' inside $\mathbf{V}$-Q_{ABCQ}. More formally we prove that for each $\varphi \in \Sigma_0^B$ and L_2-terms l, s, t there exist $\varphi' \in \Sigma_0^B$ and L_2-terms l', s', t' such that $\mathbf{V}$-Q_{ABCQ} proves

$$(\forall x < t)(M(x) \leftrightarrow \varphi(x)) \wedge (\forall x < t)(M'(x) \leftrightarrow \varphi(x))$$
$$\rightarrow (\forall X)((\exists T')ATREE(l, s, M', \langle X, c\rangle, T') \leftrightarrow OCCURS(l, s, c, M, X)).$$

The description of φ' can be easily obtained from φ. □

Now we are ready to prove the formalized version of Theorem 8.

Theorem 11 *For each* $\varphi \in \Sigma_0^B$ *and* L_2*-terms* l, s, t *there exists a function* $ACT(X)$ *which is* Σ_1^B *definable in* $\mathbf{V}$-Q_{ABCQ} *such that*

$$(\forall x < t)(M(x) \leftrightarrow \varphi(x)) \wedge (\exists T)ATREE(l, s, M, X, T) \rightarrow ATREE(l, s, M, X, ACT(X))$$

is provable in $\mathbf{V}$-Q_{ABCQ}.

(Proof). The function $ACT(X)$ can be defined as a combination of five functions which are Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.

Let T_1 be the function which lists all configurations which appears in an accepting computation of a given ATM M on input X.

Claim 1. T_1 is Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.

(Proof of Claim 1). We can define T_1 using $Q_{ABCQ}(\Sigma_0^B)$-COMP as

$$(\forall c)(T_1(l, s, M, X)(c) \leftrightarrow OCCURS(l, s, c, M, X)).$$

Let T_2 be the function which takes the output of T_1 as input and outputs the list of all pairs $\langle c, c'\rangle$ such that $c \vdash_M c'$.

Claim 2. T_2 is Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.

(Proof of Claim 2). This is easily defined by Σ_0^B-COMP as the transition relation of M is Σ_0^B definable.

Let T_3 be the function which takes the output of T_2 as input and outputs a list of pairs obtained by removing all $\langle c, c'\rangle$ except the first one for each existential configuration c of T_1.

Claim 3. T_3 is Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.

(Proof of Claim 3). Using the least number principle, we can determine whether a given pair $\langle c, c'\rangle$ is the least one for each c by a Σ_0^B relation. More formally

$$T_3(\langle c, c'\rangle) \Leftrightarrow T_2(\langle c, c'\rangle) \wedge c' = \mu_{x<|T_2|}T(c, x).$$

Such a string T_3 exists by Σ_0^B-COMP.

Let T_4 be the function which takes the output of T_3 as input and remove all configurations that are no longer reachable from the root.

Claim 4. T_4 is Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.

(Proof of Claim 4). For each configuration c appearing in T_3 execute the NL algorithm checking that it is not reachable from the root configuration. Such an algorithm is definable in $\mathbf{V}$-$Krom$ due to [3] and so is in $\mathbf{V}$-Q_{ABCQ}. Let $REACH(c, c', G)$ be the expression

of such algorithm. Since $\mathbf{V}$-Q_{ABCQ} contains $\mathbf{V}$-*Krom* it is expressed by a $Q_{ABCQ}(\Sigma_0^B)$ formula. So T_4 can be defined by $Q_{ABCQ}(\Sigma_0^B)$-COMP.

Let T_5 be the function which takes the output of T_4 as input and outputs an equivalent tree.

Claim 5. T_5 is Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.

(Proof of Claim 5). Since we know the size bound of the accepting computation tree, we rewind the DAG obtained by T_4 while simultaneously checking the size bound.

Now the function ACT can be defined as the composition of the above five functions. □

Next we turn to the formalization of Theorem 9 stating that witnesses for CSP instances can be effectively extracted from accepting computation trees of the ATM deciding them.

The ATM algorithm in [2] receives a join forest $JF(Q)$ of a given query Q as input and decide whether it is satisfied on a given database D. So in order to formalize the algorithm, we first need to check that the translation of a query to an equivalent join forest.

A join forest of a given query Q is a labeled forest (that is an acyclic undirected graph) such that

- each vertex is labeled by a conjunct of Q, and
- if two conjuncts share the same variables then the corresponding vertices are connected.

We will make use of the known algorithm for computing the minimum weight spanning forest of a given graph [6].

For a query Q let $WG(Q) = (V_Q, E_Q)$ be the Weighted Query Graph whose vertices are set of all conjuncts in Q and $(c, c') \in E_G$ whenever $c \neq c'$ and c and c' share same variables.

Proposition 3 *The function $WG(Q)$ which computes the weighted query graph is Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.*

(Proof). Use Σ_0^B-COMP. □

It is readily seen that any spanning forest is a join forest of Q. So it suffices to show that computing a spanning forest of $WG(Q)$ can be defined in $\mathbf{V}$-Q_{ABCQ}. For the construction of a spanning forest we will use the following fact:

Proposition 4 *An edge $e = \{u, v\}$ is in the minimum weight spanning forest of a graph $G = (V, E)$ if and only if it is not connected in v in the graph $G_E = (V, E_e)$ where $E_e = \{e' \in E : index(e') < index(e)\}$.*

Note that the latter condition of Proposition 4 can be checked using graph reachability.

Lemma 3 *The function $SF(Q)$ which computes the spanning forest of $WG(Q)$ is Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}. Furthermore, $\mathbf{V}$-Q_{ABCQ} proves that $SF(Q)$ is a join forest.*

(Proof). We argue informally inside $\mathbf{V}$-Q_{ABCQ}. Let the index function of edges in $WG(Q)$ be the code of edges. For each edge e of $WG(Q)$, we can compute $WG(Q)_e = \{e' : index(e') < index(e)\}$ using AC^0 function. Then use the NL algorithm to check the connectivity of e and e' in $WG(Q)_e$. Due to Kolokolova, this algorithm can be expressed by an Σ_1^B-Krom formula, say $\overline{REACH}(WG(Q)_e, e, e')$.

Finally, we can apply comprehension axiom to obtain a string T such that

$$(\forall u, v)(T(u, v) \leftrightarrow \overline{REACH}(WG(Q)_e, e, e'))$$

which proves the lemma. □

Lemma 4 *There exists a Σ_1^B definable function $ATJF(M,D,Q)$ of $\mathbf{V}$-Q_{ABCQ} such that $\mathbf{V}$-Q_{ABCQ} proves*

$$(\exists T)ATREE(l,s,M,\langle D,JF(Q)\rangle,T) \rightarrow ATREE(l,s,M,\langle D,JF(Q)\rangle,ATJF(M,D,Q)).$$

(Proof). The idea is to formalize the ATM algorithm deciding the JTREE satisfaction problem given in [2]. □

Theorem 12 *$\varphi_{ABCQ}(\Sigma_0^B)$ is self-witnessing.*

(Proof). Let $\varphi_D, \varphi_Q \in \Sigma_0^B$. We show that the existential quantifier in the Σ_1^B description of $Q_{ABCQ}[\varphi_D,\varphi_Q]$ can be witnessed by a function which is Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.

Note that φ_D and φ_Q give a database D_n and a query Q_n for each size parameter n. So we first introduce functions

$$F_{\varphi_D}(n) = D_n,\ F_{\varphi_Q}(n) = Q_n.$$

Note that these functions are Σ_1^B definable in $\mathbf{V}$-Q_{ABCQ}.

It suffices to construct a Σ_1^B definable function witnessing the existential quantifier in the formula

$$Cyclic(F_{\varphi_Q}(n)) \vee (\exists S)CQ(F_{\varphi_D}(n),F_{\varphi_Q}(n),S).$$

The first conjunct $Cyclic(F_{\varphi_Q}(n))$ contains an existential quantifier which is witnessed by a cycle in $F_{\varphi_Q}(n)$. Note that this can be reduced to the reachability problem as follows:

$$\begin{aligned} Cyclic(Q) &\Leftrightarrow (\exists a \in V_Q)Reach(a,a,Q) \\ &\Leftrightarrow (\exists a \in V_Q)(\exists P)Path(a,a,P,Q). \end{aligned}$$

The existential quantifiers in this predicate can be witnessed within $\mathbf{V}$-$Krom$ due to Kolokolova, or alternatively we can give an ATM algorithm deciding $Cyclic(Q)$ in space $O(\log n)$ and $n^{O(1)}$ tree-size and extract the witness from its accepting tree. More precisely, the algorithm works as follows:

input G
for each $a \in V_G$ *universally check* $Reach(a,a,|V_G|,G)$

procedure $Reach(a,b,n,G)$;
accept if $(a,b) \in E_G$
reject if $(a,b) \notin E_g$ *and* $n = 0$
existentially choose $c \in V_G$ *such that* $(a,c) \in E_G$
check $Reach(b,c,n-1,G)$

Each configuration of the ATM on the execution of $Reach(a,b,n,g)$ contains the pair $\langle a,b,n\rangle$. So it is readily seen that this algorithm requires $O(\log n)$ space.

It is also easy to see that it computes $Cyclic(Q)$ and has polynomial tree-size. We claim that any path in an accepting computation tree of this ATM on input G contains a list of vertices which form a cycle in G.

Now applying the formalized version of the witnessing of ATMs, we obtain a Σ_1^B definable function

$$Cycle(Q) \rightarrow Path(CY(Q)[0],CY(Q)[0],CY(Q),Q).$$

Finally, the self witnessing property of $Q_{ABCQ}(\Sigma_0^B)$ is guaranteed by the combination of two functions $ATJF$ and CY. □

5 Future Works

We end the paper by stating an on-going work relating our result.

One of the major problems to go beyond our work is the construction of a theory for LOGDCFL. A possible idea for this problem is to use tree-size characterizations

Theorem 13 (McKenzie et.al.[4]) *LOGDCFL is the class of predicates which are computed by polynomial size multiplex circuits having polynomial size proof trees.*

An idea for the theory based on this characterization is to formalize the following type of statement:

> Given a circuit C and an input X, we can decide whether a gate in C has a polynomial size proof tree on input X.

More precisely, we can write an L_2-formula describing the following statement:

> Let C be a multiplex circuit and X be an input , there exists a string Z such that $g \in Z$ if and only if g is a gate in C which has a polynomial size proof tree on input X.

By extending $\mathbf{V}^0$ by the axiom as above, we may obtain a theory for LOGDCFL.

References

[1] G. Gottlob, N. Leone, and F. Scarcello. The complexity of acyclic conjunctive queries. Jounal of the ACM, 48(3), pp.431–498, (2001)

[2] G. Gottlob, N. Leone, and F. Scarcello. Computing LOGCFL Certificates. Theoretical Computer Science, 270(1-2), pp.761-777, (2002)

[3] A. Kolokolova, Systems of Bounded Arithmetic from Descriptive Complexity. PhD dissertation, Toronto University, (2005)

[4] P. McKenzie, K. Reinhardt and V. Vinay, Circuits and Context-Free Languages. COCOON 1999,pp.194-203, (1999)

[5] P. Nguyen, Bounded Reverse Mathematics, PhD dissertation, Toronto University, (2008)

[6] J.H.Reif. Symmetric complementation. In Proc. 14th ACM Symposium on Theory of Computing, pp.201–214, (1982).

[7] H. Vollmer, Introduction to Circuit Complexity. Springer (1999).